

# プロジェクトマネージャ育成シミュレータにおける 決定木を用いた学習者オペレーションの改善方式

An Improvement Method of User Operations using Decision Tree  
on Project Manager Skill-Up Simulator

大月 みなみ\*<sup>1</sup>

Minami OTSUKI

秋吉 政徳\*<sup>2</sup>

Masanori AKIYOSHI

鮫島 正樹\*<sup>1</sup>

Masaki SAMEJIMA

\*<sup>1</sup>大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

\*<sup>2</sup>広島工業大学情報学部

Faculty of Applied Information Science, Hiroshima Institute of Technology

Project managers of software development need to perform various operations as overtime directive or supervising action. We have developed a simulator in order to improve such skills, but the simulator does not provide a feedback of operations. In order to evaluate user operations, our proposed method generates various operations by agent programs and builds a decision tree to associate a project result with operations. Then improved operation point is generated by the decision tree and a user's operation log. Finally improved operations that lead to better project result are induced.

## 1. 序論

近年、システム開発プロジェクトは複雑化・多様化しており、プロジェクトを成功裡に終わらせることは困難になっている。そのため、プロジェクトを成功に導くことができる優秀なプロジェクトマネージャ(以下、PM)が必要とされている。PMはあらかじめPMBOK[1]からプロジェクトマネジメントの知識体系を学び、実際のプロジェクトに知識を生かすことにより経験を積む。従来は、OJT(On-the-Job Training)、ケーススタディ[2]、ロールプレイング[3]によって教育されることが一般的である。しかし、指導者への大きな負担や時間がかかってしまうこと、さらにOJTでは実際のプロジェクトを育成段階のPMに任せることによる失敗のリスクといったデメリットから、多くのプロジェクトに参加させることが困難である。

そこで、プロジェクトを疑似体験できる学習環境を構築することでPM育成を支援するために、PM育成シミュレータの研究が行われている[4, 5]。シミュレータを用いて学習を行うことで、コストをかけずに様々なプロジェクトを疑似体験でき、失敗によるリスクもないという利点がある。PMの育成を支援するために筆者らが研究開発を行っているシミュレータは、プロジェクトマネジメントにおける実装管理フェーズを模擬する対話的な学習環境である。指導者が設定したプロジェクトの特性をもとに、日々の進捗具合やバグの発生、あるいは外乱的なイベントにより作業効率が落ちるといったことを模擬し、学習者はそれらの状況に応じて自らの判断でオペレーションを行うことで、プロジェクトマネジメントを学ぶことになる。しかしながら、シミュレータでは単にプロジェクトを疑似体験できるのみであり、学習者に対する学ぶべきことへのフィードバックが行われず学習環境として不十分である。

本稿では、現在開発中のシミュレータにおいて、フィードバックとして学習者のオペレーションにおける改善箇所を提示する学習環境の構築を目的とし、学習者オペレーションの改善方式を提案する。

## 2. 学習者オペレーションの改善方式

### 2.1 育成シミュレータの概要と課題

図1に、対象とするプロジェクトマネージャ育成シミュレータの構成を示す。指導者が設定したプロジェクトを構成する各作業の技術ドメインとその難易度や作業間の親子関係、作業を行う際に要求される人員のスキルレベル、またプロジェクトの状況に応じて発生するバグの追加や工数追加といったイベントが、プロジェクトモデルやイベントルール群として保存されている。これらをもとにシミュレータが動作し、学習者はガントチャートから各作業の日々の進捗状況、進捗チェックの際にはバグの検出による作業遅延の確認、残業やより高いスキルレベルを持つ人員(以下、熟練者)との共同作業といったオペレーションを作業ごとに指示することができる。プロジェクト終了後にはプロジェクト結果とともに、その評価として一般的に用いられる品質・コスト・納期を意味するQCD(Quality, Cost, Delivery)が提示される。ここでは、Qの指標を総検出バグ数、Cの指標をオペレーション実施による総追加コスト、Dの指標をプロジェクト期間と定義し、すべて値が小さいほど結果が良いとする。

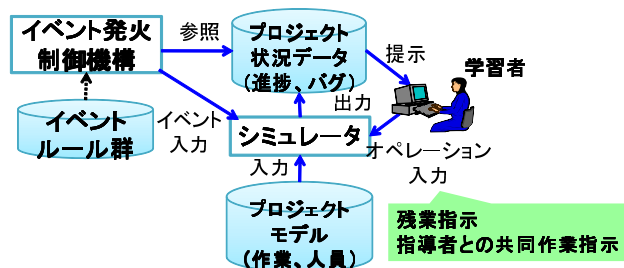


図1: プロジェクトマネージャ育成シミュレータの構成

学習者は各作業に対して様々なオペレーションを指示することによってプロジェクト状況を変化させ、より良いプロジェクト結果を得ようとする。プロジェクト結果が悪い場合、学習者による一連のオペレーションの中で不適切なオペレーションがいくつかあると考えられるが、その中でもオペレーションを変

連絡先: 大月みなみ、大阪大学大学院 情報科学研究科 マルチメディア工学専攻 ビジネス情報システム講座、〒565-0871 大阪府吹田市山田丘 2-1、otsuki.minami@ist.osaka-u.ac.jp

更することで最も良いプロジェクト結果が得られる改善箇所について、学習者に提示することが、学習効果を高めるうえで必要である。しかし、改善としては、学習者がオペレーションを行っていない箇所に対してもオペレーションを行うべき場合も考慮しなければならないが、学習者のオペレーションを一つづつ異なるオペレーションに変更することで改善箇所の特定を行うことは、現実的でない。そこで、プロジェクト結果との関連付けを考案し、その関連付けを利用したオペレーションを抽出することによって改善箇所の特定を行う。

## 2.2 学習者オペレーションの改善方式の概要

提案方式の概要を図2に示す。まず、学習者がシミュレータを用いてプロジェクトに関するオペレーションを実行し、その際のプロジェクト状況に依存して発生するイベントをすべて記録する。学習者によるプロジェクト終了後、同一イベントを発生させることで学習者が体験した状況と同じ状況を作り、エージェントプログラムを用いて様々なオペレーションを適用した場合のプロジェクト結果をエージェントログとして得る。学習者とエージェントプログラム両方のオペレーションとプロジェクト結果から、オペレーション特徴抽出機構内で決定木を作成し、オペレーションとプロジェクト結果を関連付ける。

学習者のプロジェクト結果よりも改善されるオペレーションを作成し単純にマッチングすることで改善箇所を特定するには、図3の矢印で表されるオペレーション実施箇所が示すようにオペレーション内容と実施日が一致する場合はほぼないことから、困難である。従って、図3の破線のように各作業期間をある日数ごとに区切った場合の各期間内におけるオペレーション回数をオペレーション特徴と定義し、日付ごとのオペレーションのマッチングではなくオペレーション特徴を用いた改善箇所の特定を行う。

改善箇所特定機構では、学習者のプロジェクト結果に至る要因となったオペレーション特徴を変更することで、より良いプロジェクト結果が得られるならば、その場合のオペレーション特徴を基に改善オペレーションを作成し、シミュレータを用いてプロジェクト結果を得る。最も良いプロジェクト結果が得られたオペレーション特徴に対応した学習者オペレーションの箇所を改善箇所とする。

## 2.3 プロジェクト結果の条件となるオペレーション特徴の抽出

オペレーション特徴の抽出を行うために、決定木を利用する。決定木は葉ノードを結果とし、枝が結果に至るまでの特徴の集まりを表すような木構造となっている。作成した決定木から、学習者のオペレーション特徴群をもとに学習者より良いプロジェクト結果となるオペレーション特徴群を抽出する。

### 2.3.1 プロジェクト結果の評価値算出

決定木を作成するためには結果の良し悪しに応じて決定する必要があることから、プロジェクト結果を評価するための指標を作成する。プロジェクト結果は、本来であれば熟練したPMによる評価が必要であるが、エージェントプログラムを用いて作成された多数のプロジェクト結果をすべて評価してもらうことは不可能である。従って、プロジェクト結果を評価する指標を、以下のように考える。

評価対象は、QCDのQの指標である総検出バグ数とDの指標であるプロジェクト期間とする。プロジェクトにおけるバグは作業担当者が検出できた検出バグと検出しきれずに残ったバグである潜在バグの2種類があるとし、両者を合わせて発生バグとする。本シミュレータでは進捗チェックの際に十分にバグの検出が行われることから、検出バグ数は発生バグ数と正

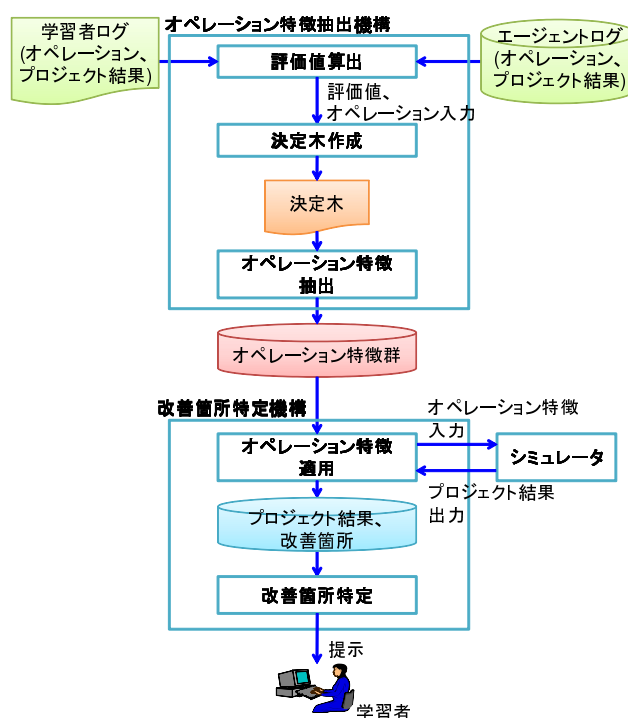


図2: 学習者オペレーションの改善方式

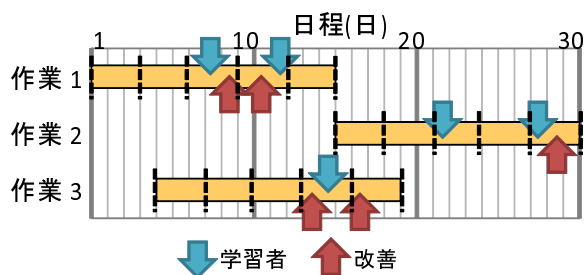


図3: オペレーション実施箇所

の相関があることを前提とし、検出バグ数が少ないほど発生バグ数も少ないと考え、品質が良いとする。また、学習者がオペレーションを実施することで費やした総追加コストよりも高いコストを費やすことでQとDを改善するのではなく、学習者が実施したオペレーションによるコストの範囲内でQとDを改善するオペレーションを作成する。このため、後述する決定木作成の際には学習者が費やしたコスト以下のデータのみを使用することから、ここではCである総追加コストを評価対象に含めない。評価はn件のプロジェクト結果の相対評価とする。

n件のうち総検出バグ数とプロジェクト期間の最小値をそれぞれ $Q_{min}, D_{min}$ 、最大値を $Q_{max}, D_{max}$ とする。あるプロジェクト結果 $P_i (i = 0, \dots, n)$ の評価値を得る場合、まず $P_i$ における総検出バグ数 $Q_i$ とプロジェクト期間 $D_i$ を、式(1),(2)を用いて各々0から1の値に正規化する。正規化後の値をそれぞれ $Q'_i, D'_i$ とする。

$$Q'_i = \frac{Q_i - Q_{min}}{Q_{max} - Q_{min}} \quad (1)$$

$$D'_i = \frac{D_i - D_{min}}{D_{max} - D_{min}} \quad (2)$$

次に、式 (3) により算出された  $evaluation_i$  から、同様に式 (4) を用いて 0 から 1 に正規化した値  $evaluation'_i$  を算出する。式 (5) に基づいて、 $evaluation'_i$  から 5 段階評価値  $fivelevel_i$  を得る。値が大きいほど、評価が高いとし、5 が最も高い評価を示す。

$$evaluation_i = Q'_i + D'_i \quad (3)$$

$$evaluation'_i = \frac{evaluation_i - evaluation_{min}}{evaluation_{max} - evaluation_{min}} \quad (4)$$

$$fivelevel_i = \begin{cases} 5 & (0.0 \leq evaluation'_i < 0.2) \\ 4 & (0.2 \leq evaluation'_i < 0.4) \\ 3 & (0.4 \leq evaluation'_i < 0.6) \\ 2 & (0.6 \leq evaluation'_i < 0.8) \\ 1 & (0.8 \leq evaluation'_i < 1.0) \end{cases} \quad (5)$$

### 2.3.2 オペレーション特徴を表す決定木の作成

プロジェクト期間中は、各作業に対して様々なオペレーションが行われるが、その中でも特にプロジェクト結果を左右するいくつかのオペレーション特徴があると考えられる。改善箇所の特定のためには、学習者のプロジェクト結果の要因となったオペレーション特徴やより良いプロジェクト結果と関連付けられるオペレーション特徴を抽出することが重要となる。そこで、オペレーション特徴とプロジェクト結果から算出される 5 段階評価値から決定木を作成する。

決定木作成には入力として多数のデータが必要となることから、様々なオペレーションを行うエージェントプログラムとして以下の 5 つのタイプを用いる。

- 規範的  
遅延時に残業指示、担当人員のスキルより作業難易度が高い場合に熟練者との共同作業を指示
- 残業過多  
残業を頻繁に指示
- 熟練者との共同作業指示過多  
熟練者との共同作業を頻繁に指示
- オペレーション遅れ  
作業初期にはほぼオペレーションなし
- ランダム  
残業、熟練者との共同作業を確率的に指示

エージェントプログラムをプロジェクト中の各作業に割り当てること、あるいは一つのプロジェクトで一貫して同じエージェントプログラムを割り当てることにより、様々なオペレーションによるプロジェクト結果を取得することが可能となる。

決定木の属性はオペレーション特徴とし、残業指示、熟練者との共同作業指示、積極的に指示しなかった場合をそれぞれ作業開始日からある一定期間ごとに区切った場合の期間内に実施した回数とする。「積極的に指示しなかった」とは、進捗チェックを行った際、予定よりも作業進捗が遅い場合や想定以上のバグの発生による遅延が発覚した場合に、敢えて何も指示をしないで問題ないと判断した場合を意味する。属性として「指示なし回数」や「指示回数」を用いることで、学習者によるオペレーションの未実施に関わる改善箇所の特定が可能となる。ま

た、クラスは評価方式により算出した 5 段階評価値とする。決定木作成の入力データの条件として、学習者がオペレーションを実施することにより費やした総追加コスト以下とする。この条件により、学習者による総追加コストの範囲内で指示可能なオペレーションを行ったデータのみを収集できることにより、QCD における C の指標である総追加コストは必ず学習者よりも改善されていることが保証される。決定木の例の一部を図 4 に示す。

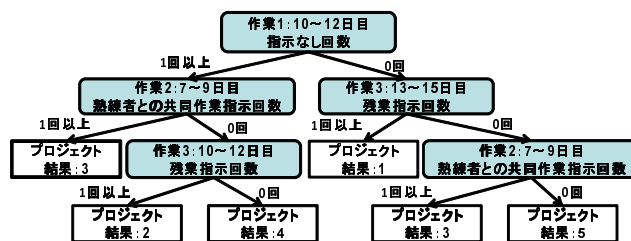


図 4: 決定木の例

### 2.3.3 オペレーション特徴の抽出

決定木を作成することで、プロジェクト結果とオペレーション特徴を関連づけることができる。決定木上で、学習者が行ったオペレーションに当てはまる枝をたどることにより、最終的にはある葉ノードにたどり着き、そのノードが示す評価値が学習者の評価値となる。葉ノードに至るまでのノードの数だけ結果と関連づけられる学習者のオペレーション特徴があることから、学習者に当てはまるオペレーション特徴をまとめて学習者のオペレーション特徴群と呼ぶ。

### 2.4 改善箇所の特定

学習者のオペレーション特徴群の中から 1 つを選択し、学習者に当てはまる枝とは異なる枝を辿ることで、学習者のプロジェクト結果の評価値よりも良い評価値が得られる経路を探索する。例えば、改善箇所を特定するための経路選択の一例を示す図 5 では、根ノードとその左の子ノードに対応する枝が学習者のオペレーション特徴であることから、2 つのノードの中からまず根ノードを選択したとする。その場合、学習者とは異なる値を取る枝を辿り、最終的に学習者の評価値 2 よりも良い評価値が得られる経路を探索する。図 5 では破線の矢印で示される経路が学習者よりも良い評価値が得られる経路であることから、この経路を構成する枝がより良い評価値を得るためのオペレーション特徴を示している。探索により得られたオペレーション特徴を基に改善オペレーションを作成する。

改善オペレーションは最も良い結果が得られる改善箇所以降のオペレーションとし、改善箇所以前は学習者のオペレーションが適切であったと考える。そのため、より良い評価値が得られる経路上のオペレーション特徴において、改善箇所以前の特徴が表れた場合は学習者に当てはまる枝を必ずたどることとする。決定木はプロジェクト結果と関連づけられるオペレーション特徴のみを示すことから、経路上に現れないオペレーション特徴もある。従って、改善オペレーション作成時に、決定木の経路上に現れたオペレーション特徴以外でオペレーション指示が可能な場合はランダムにオペレーションを適用し、複数回シミュレーションを繰り返す。

以上を学習者の全てのオペレーション特徴について繰り返し行い、最もプロジェクト結果が良い改善オペレーションに含まれるオペレーション特徴を基に、該当する学習者オペレーションの改善箇所を特定する。図 5 の決定木で繰り返し上記の操作を行ったとし、最もプロジェクト結果が良い改善オペレーシ

ンを構成する経路が破線の矢印で示される経路である場合は、「作業1の10日目から12日目に指示を行わなかったこと」が改善箇所となる。学習者には改善箇所、及び改善箇所以降のオペレーション例を提示する。

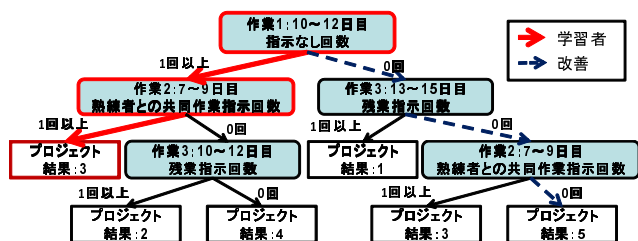


図 5: 改善オペレーション特徴群の探索例

また、改善箇所以降のオペレーションを適用した場合と学習者によるプロジェクト結果を図8に示す。プロジェクト結果は総検出バグ数に基づく品質(Q)、オペレーション実施による追加コスト(C)、プロジェクト期間に基づく納期(D)である納期とする。QCDすべてにおいて、学習者のプロジェクト結果よりも改善されていることが分かる。

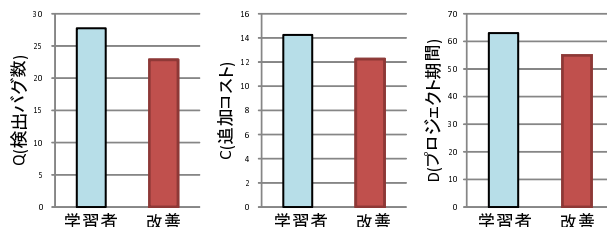


図 8: プロジェクト結果

### 3. 評価・考察

#### 3.1 実験

実験対象として、作業1、2、3と作業4、5がそれぞれ依存関係にあり、作業1、2、3がクリティカルパスとなるようなプロジェクトモデルを用いた。また、作業1、3、4は担当人員のスキルレベルが低く、遅延することが予想されるようなモデルとした。なお、決定木を構築するために、オープンソースのデータマイニングツールである WEKA に含まれる C4.5 アルゴリズムを実装した J48 を用いた [7]。

学習者によるオペレーションを図6に示す。

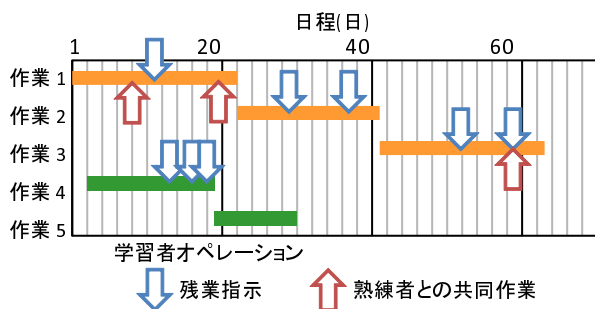


図 6: 学習者のオペレーション

#### 3.2 実験結果

特定した改善箇所と改善箇所以降のオペレーションを図7に示す。改善箇所(丸印)として、作業1の15日目に熟練者との共同作業指示を実際に行うべきだったが、未実施であることが指摘された。

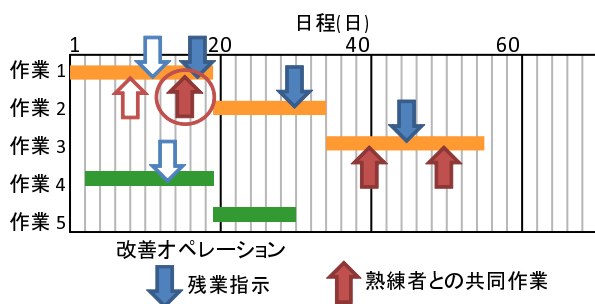


図 7: 改善箇所と改善オペレーション

### 4. 結論

本稿では、プロジェクトマネージャ育成シミュレータにおける学習者オペレーションの改善方法を提案した。学習者によるオペレーションの改善箇所を特定するには、実施したオペレーションだけでなく、未実施のオペレーション箇所についても考慮する必要があり、人手で特定することは困難である。提案方式では、決定木を用いて、学習者のプロジェクト結果と関連付けられるオペレーション特徴群、さらに良いプロジェクト結果の要因となるオペレーション特徴を把握することで改善箇所を特定し、提示することができた。

### 参考文献

- [1] A Guide To The Project Management Body Of Knowledge (PMBOK Guides), Project Management Institute, 2004.
- [2] 井沢, 鳥山見, 杉田, “ 実践ケーススタディ研修によるプロジェクトマネージャの育成, ”プロジェクトマネジメント学会研究発表大会予稿集, pp.233-235, 2003.
- [3] T. R. Henry and J. LaFrance, Integrating Role-play into Software Engineering Courses. J. Comput. Sci. Coll., Vol.22, No.2, pp.32-38, 2006.
- [4] A. Drappa and J. Ludewig, Simulation in Software Engineering Training, in Proc. of the 2000 Int. Conf. on Software Engineering, pp.199-208, 2000
- [5] K. Iwai, M. Akiyoshi, M. Samejima, and H. Morihisa, A Situation-dependent Scenario Generation Framework for Project Management Skill-up Simulator, In Proc. of the 6th Int. Conf. on Software and Data Technologies, Vol.2, pp.408-412, 2011.
- [6] A. J. G. Babu and N. Suresh, Project Management with Time, Cost, and Quality Considerations, European Journal of Operational Research, Vol.88, No.2, pp.320-327, 1996.
- [7] <http://www.cs.waikato.ac.nz/ml/weka/>